Song Lyric Analysis

Iona Buchanan

April 24, 2018

MAT 4376 E

Contents

1 Introduction

A song has two main components: the music and the lyrics (sometimes lack thereof). The purpose of this project is to identify patterns and trends in song lyrics. We also look to classify songs by genre or by topic using their lyrics. From this analysis, we hope to gain insight into the music industry and pop-culture which could influence song-writing and playlist creation.

1.1 Outline and Objectives

Collection The sample of songs was downloaded from a Kaggle. Additional metadata was then web-scraped for each song. This process can be very time consuming and computationally expensive.

Exploration Preliminary exploration was conducted to identify the variety of songs and trends in our sample. With a representative dataset, it is possible to draw parallels between trends in the music industry and trends in our data.

Genre classification We attempt to use a text embedding model, *Word2vec*, to classify each observation by genre. Although this models is not new, we provide an original application and determine its effectiveness.

Topic modelling Text embedding models also allow the identification of the topic of a document. We used this dataset to demonstrate this technique, as well as the ability to search for lyrics pertaining to a specific topic. This task can be supervised or unsupervised and the results can be used for many purposes such as creating playlists based on topic.

2 Data

There are many public websites with lyrics and information about songs for personal interest (eg. Genius.com, AZlyrics.com, metrolyrics etc). However, since the distribution of lyrics can pose legal and copyright issues, it is not always easy to obtain large amounts of this data, even for personal research purposes.

2.1 Collection

For this project, a data-set of 56 550 songs was collected, along with the song title, artist, lyrics, release year and genre for each observation. I began with a large data-set from Kaggle.com of over 350 000 songs. However, after some exploratory analysis and validation using some of the websites mentioned above, the data was found to be very messy and had too many collection errors. We therefore settled for a smaller but more representative data-set, which has lyrics

for each song but not the release date or genre. We then used Genius API and general webscraping to retreive the required metadata for each observation.

2.2 Cleaning

Not all searches were successful, so there were a number of songs missing a genre and/or release date. We scraped genre and album names for about two thirds of the dataset. Since the lyrics are the primary feature of the dataset, and the other features are only used for exploratory analysis and validation, we will accept a large proportion of missing values and assume they are randomly distributed.

3 Exploratory Analysis

We did not choose the songs included in the dataset, so it was important to assure the sample would represent the music industry accurately.

3.1 Basic descriptive statistics

Genre This was one of the features that was not included in the original dataset and will vary depending on the web-scraping source. Genre is defined to be the "category of [musical] composition characterized by a particular style, form, or content"[2]. Genre has multiple levels and is in some sense, dynamic, with new genres being born out of old ones. This makes tagging genre partly subjective. Pop and Rock are one of the most common genres in our dataset which is understandable since any other genres are thought to be sub-genres of pop, rock, country, hip-hop/rap and R&B.

Release dates Ninety percent of the songs in the dataset were released after 1970. Figure 1 shows the lack of data for the time period prior to 1970. We therefore are not able to make conclusions for these years with confidence. We do see that the distribution of genres is relatively the same for each year, which allows us to conclude trends with respect to genre over time.

Genres and song length A distinctive feature of rap is that the lyrics are spoken rhythmically over a beat which causes them to be uttered faster. We therefore expect the word-counts for rap songs to be relatively higher than for other genres. In contrast, jazz and classical music are primarily instrumental forms of music and thus will have less words [3]. We see this is true of our data-set in Figure 2.

3.2 Complexity and Repetition

One aspect of music that has changed is the complexity or repetitiveness of lyrics. Gao, Harden et al. suggest that repetitive lyrics are more catchy and

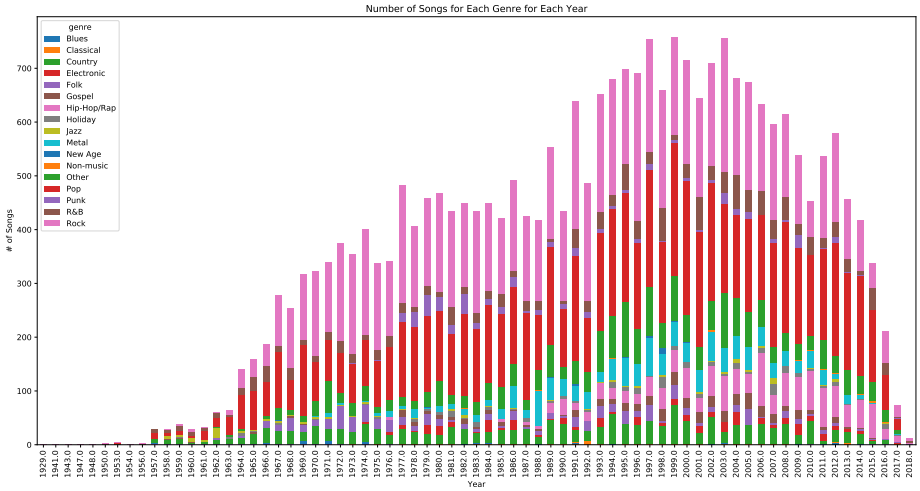

Figure 1: Distribution of genres by year

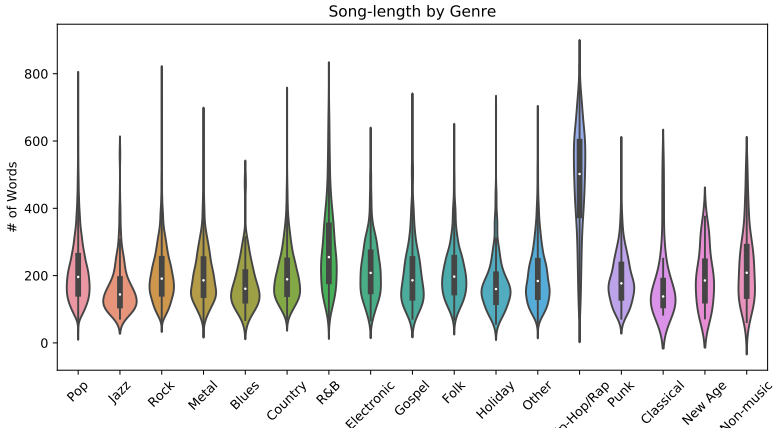

Figure 2: Distribution of song-length by genre

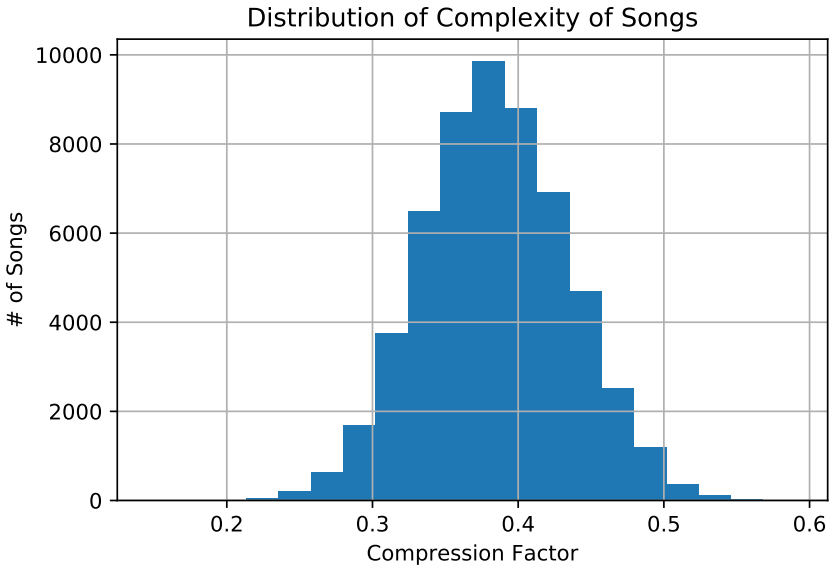

Figure 3: Distribution of complexity across the dataset

thus become more popular [4]. But how do we go about measuring repetitiveness? One metric, used by Colin Moris in an online visual essay, relates compressablility to complexity [5]. The Lempel-Ziv algorithm is a data compression algorithm which consists of compressing repeated strings.[6] We therefore define the Lempel-Ziv complexity as

$$\frac{l(C(x))}{l(x)}$$

where $l(C(x))$ is the length of the compressed data and $l(x)$ is the length of the original data. A highly repetitive song would therefore be more compressed and thus have lower complexity.

Figure 3 shows that the complexity factors for the lyrics in our sample are normally distributed with mean 0.4. In other words, most songs can be compressed by about 60%. The high compression rate (relative to regular text which compresses about 8%) is mostly due to the rhyming structure of lyrics. Figure 4 shows the decrease in complexity of songs over the years which could explain a recent relationship between complexity and popularity. We also find that electronic, R&B and rap songs are the most repetitive genres (Figure 5).

3.3 Profanity

Another aspect of music that has changed over time is the use of profane language. In general, it has increase over time and has evolved with the rest of

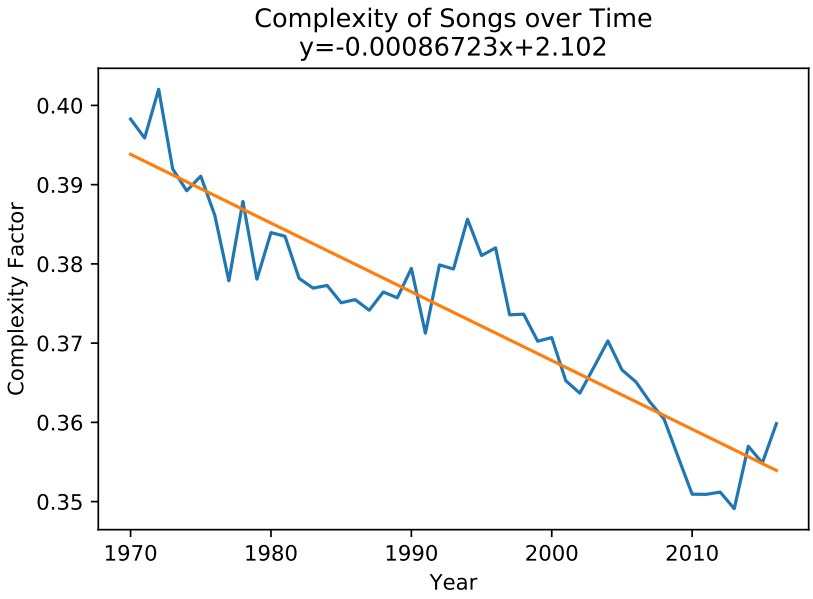

Figure 4: Distribution of genres. Given the lack of data prior to 1970, only data between 1970 and 2017 is shown

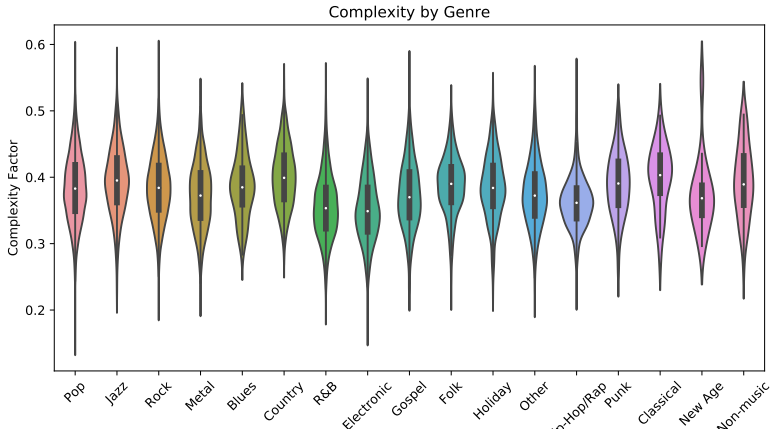

Figure 5: The complexity factor for songs of each genre

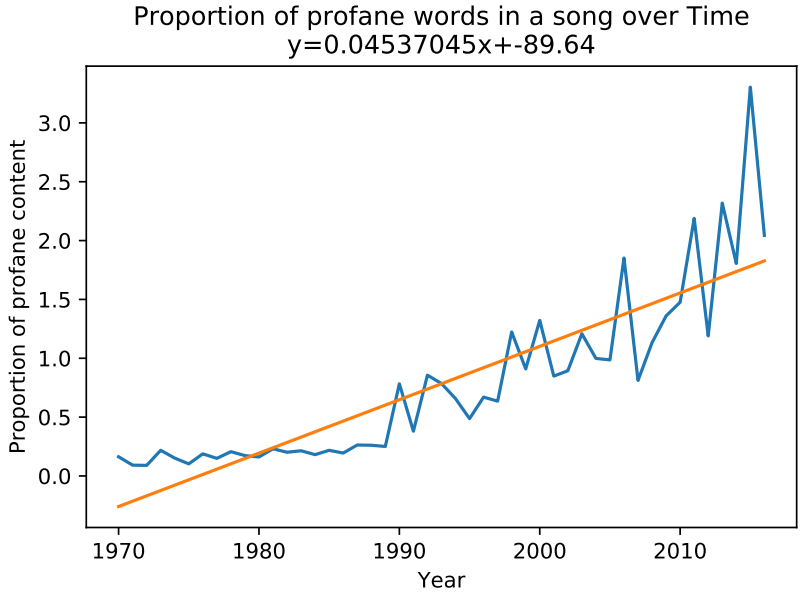

Figure 6: Increase in Profanity over Time

colloquial language. (Figure 6) A dictionary of profane words was collected from [7] and a document-term matrix was created for the lyrics in the dataset.

Slang (including profane slang) changes with society over time. Common curse words of today are not necessarily the words of choice from previous decades. The occurrence of swearing also differs from one genre to another, with rap having by far the most profanity (Figure 7).

4 Genre Classification

One of the main goals of this project is to identify the genre of a song using the lyrics. "The traits that define a genre are more than a similar sound or summary of technical elements; subculture, fashion, geography, mentality and period of time all qualify as possible trademarks of which a genre in retrospect might be recognized." [8]. Some of these aspects appear in the lyrics and we would like to see if the lyrics alone are enough to estimate the genre of a song. We propose two different methods: embedding the lyric text, and modelling using features of the lyrics such as word-count and complexity.

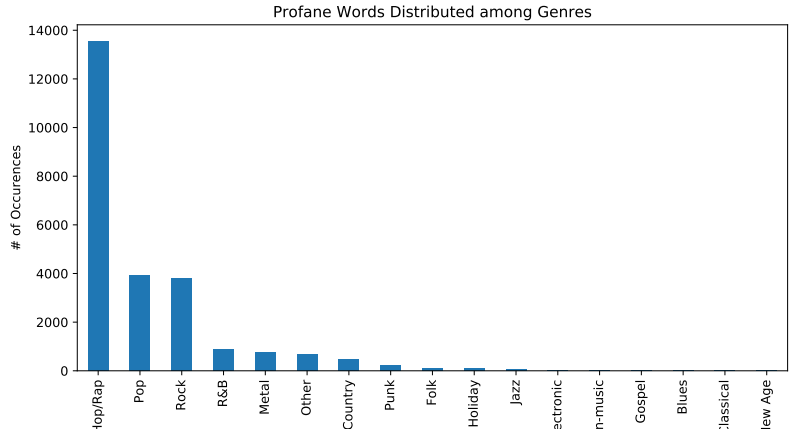

Figure 7: Profanity by Genre

4.1 Lyric2vec

4.1.1 Word2vec

Word2vec Our first method uses a technique called *Word2vec*. Word2vec is a word embedding technique created by Google [9] which maps words in a given training corpus to vectors in a high-dimensional vector space (100 by default). The aim is to embed words that are similar in a conceptual sense to vectors that are mathematically close to each other in the resulting space [10]. For example, "car" and "truck" would be close to each other because they are both vehicles, but each far away from "banana" since it is a fruit. It also represents analogies: "king" - "man" + "woman" = "queen".

Similarity In the context of Word2vec models, similarity score between two vectors $\vec{a}$ and $\vec{b}$ is taken to be

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

where θ is the angle between $\vec{a}$ and $\vec{b}$ [11]. Since $\cos\theta$ takes values between -1 and 1, the maximum similarity score is 1, which occurs when $\vec{a} = \vec{b}$. In other words, a word is always most similar with itself.

doc2vec We can now perform what is called *doc2vec*. This embeds documents (in our case, lyrics) into the word vector space such that each document is the average of its word vectors. This way, we are able to maintain the relationship between conceptual similarity and distance. Lyrics that speak of similar things should be close to each-other in the vector space. This model is implemented as a part of *gensim* [12].

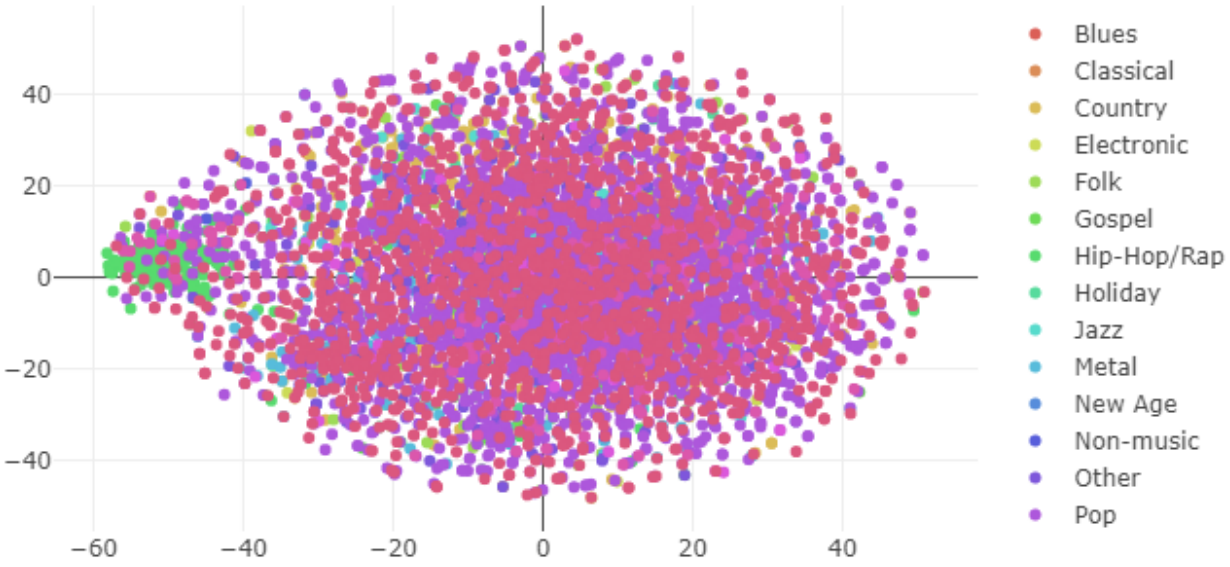

Figure 8: 100-dimensional embedding of lyrics, reduced to 2D for visualisation purposes. Each point represents the lyrics to a song and the colour represents the genre.

Figure 8 is a two-dimensional visualisation of the embedding, colour-coded by genre. The reduction was performed using t-SNE (t-Distributed Stochastic Neighbor Embedding), a dimension reduction technique which is well-suited for high-dimensional datasets [13]. The analysis is done on the 100-dimensional vectors and this reduction is only for the purposes of visualisation. Although high-dimensional data does not cluster very well, we can see that rap songs are relatively close to each-other. This indicates that rap has a distinctive vocabulary.

Genre Vectors In order to estimate genres, we will create *genre vectors*. These vectors will be calculated using words that occur frequently in a particular genre. Thus, lyrics that are close to a given genre vector are estimated to be of that genre.

4.1.2 Results

The classification was attempted using two different sets of genre vectors: one created using the dataset, and another based off the word clouds from a paper, *Lyrics-based Analysis and Classification of Music*, by Fell and Sporleder [15]. For each song, the cosine similarity to each genre vector is calculated and the most probable genres for a given song are estimated to be the ones with the highest similarity. Since genres are often difficult to determine and there are not clear cut definitions, we estimated the three most likely genres for a given song and deem a prediction successful if the pre-tagged genre is one of the three estimated genres.

This dataset The genre vectors were built from a *tf-idf* (term-frequency inverse-document-frequency) matrix. The complete lyric set for each genre was treated as a document in order to identify terms that are specific to each genre. We describe each genre with the 100 terms for each genre with the highest tf-idf score. For example, the term "love" appears in almost all genres and therefore is not a good descriptor. However, "Christmas" appears often and almost exclusively in holiday songs so this is a good descriptor of the holiday genre.

Outcome Figure 9 shows the number of times a song of a particular genre (vertical) was estimated to be of another (horizontal). The model showed bias towards Jazz and R&B which may mean that the topic vectors for these genres were too general. This model had a success rate of about 20%. Under normal standards, this is not a good model but it gives an idea of how much the lyrics contribute to the genre of a song. We also note, that we have only trained these genre vectors on our sample of 55 000. We, therefore, repeat with genre vectors trained on a larger sample.

Word clouds In Fell and Sporleder's paper, they analyse a dataset of over 400 000 lyrics (compared to our 55 000) so we hope to get a more accurate

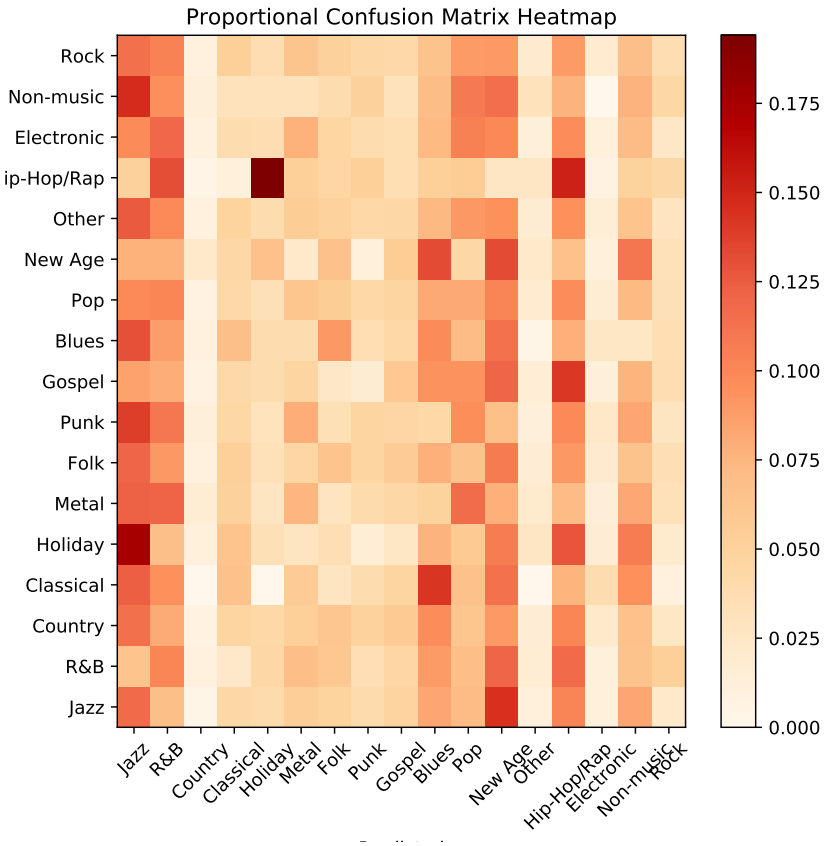

Figure 9: Proportional convolution matrix heatmap for topic vector model

prediction. These genre vectors are then the average of the words in each word cloud (Figure 10). One issue when it comes to validating this model is that the genres identified by Fell and Sporleder are not necessarily the same ones that we identified in our dataset. Therefore, we do not have genre vectors for the missing genres such as "Holiday" and "Electronic".

Outcome These vectors performed slightly worse that the ones created using the dataset but are less prone to over-fitting. Between the three guesses for each song, the model was able to identify the actual genre about 30% of the time. Figure 11 is a confusion matrix standardized by the total number of actual tags of each genre. We see that often the model would falsely tag songs as being of the genre R&B, Country or Gospel. This indicates that perhaps the genre vectors for these genres could be refined.

Evaluation We therefore conclude that genre is represented in some aspects by the words of a song but that this alone is not enough to confidently classify lyrics by genre. Although the performance was not ideal, it can possibly be improved by refining the genre vectors. Multiple sources can be combined to produce better vectors. The advantage to this method is that the definition of genres does not depend on the dataset and can be created from external sources with no knowledge of the testing set.

4.2 Modelling lyric features

We now consider a second method of embedding. Instead of embedding the raw or tokenized lyrics, we will embed features of these lyrics such as length, complexity, language and amount of profanity. Assuming all variables are categorical, we can run algorithms such as decision trees to predict genres. If there are categorical variables, we can encode them using numerical mappings or string hashing. The features used were:

- Song length
- Complexity factor
- Language
- Profanity proportion
- 1st person references (I, me, my, ...)
- 2nd person references (you, your, ...)
- Male references (he, his, ...)
- Female references (she, her, ...)

Figure 3: Blues top 100 words

Figure 4: Rap top 100 words

Figure 5: Metal top 100 words

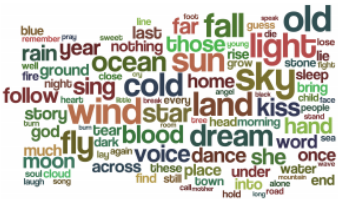

Figure 6: Folk top 100 words

Figure 7: R&B top 100 words

Figure 8: Reggae top 100 words

Figure 9: Country top 100 words

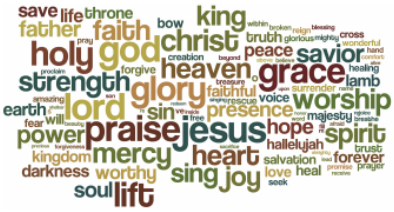

Figure 10: Religious top 100 words

Figure 10: The genre word clouds from *Lyrics-based Analysis and Classification of Music*, by Fell and Sporleder [15]

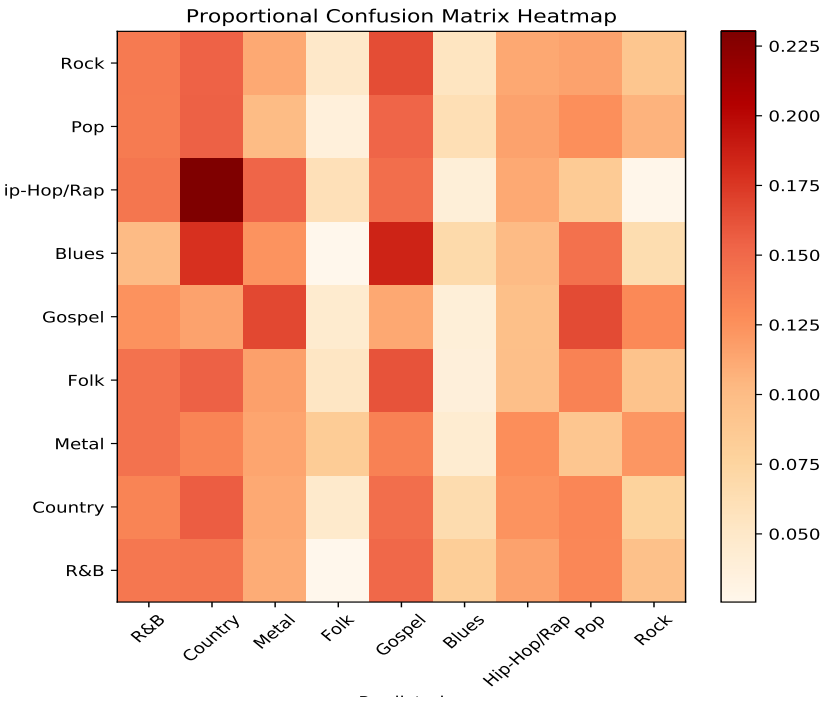

Figure 11: Proportional confusion matrix heatmap for word cloud model

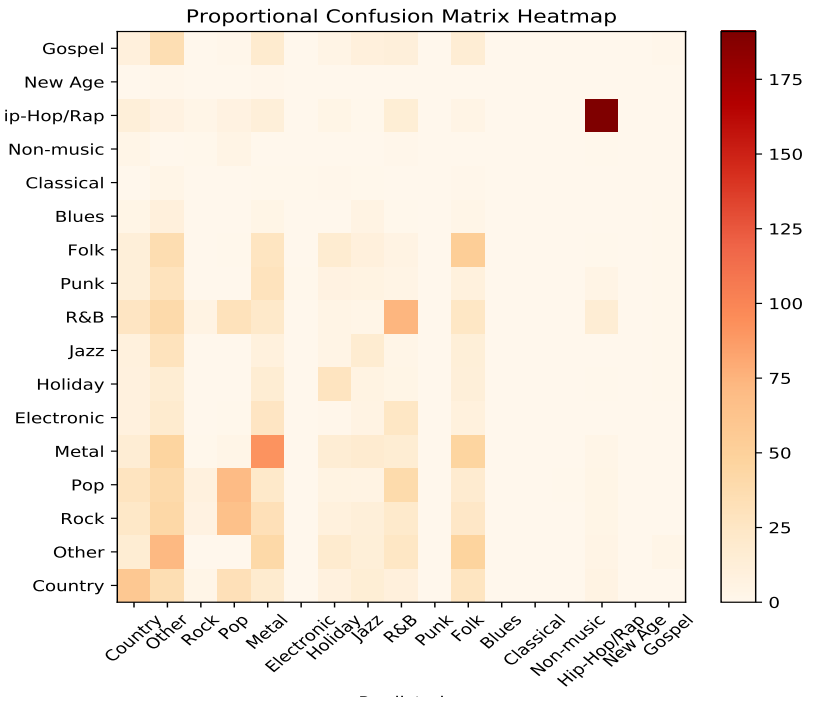

Figure 12: Confusion matrix heatmap for feature model

4.2.1 Results

These results were not very good either. The model predicted genre with an accuracy score of about 0.27. However, figure 12 shows that this model does not exhibit the bias that the Word2vec prediction model suffered from. It also shows that rap is a very distinct genre when considering the lyrics.

4.3 Combining both models

Since the word2vec and feature models represent different aspects of the data, we hoped to get a better prediction by combining the two. This new model with include the features used previously, as well as the similarity score to each genre from the word2vec model. This model actually performed worse than both individual models at 25% accuracy. It is possible that the combined model suffered from over fitting. Future work would therefore aim to find a more accurate combination of these models.

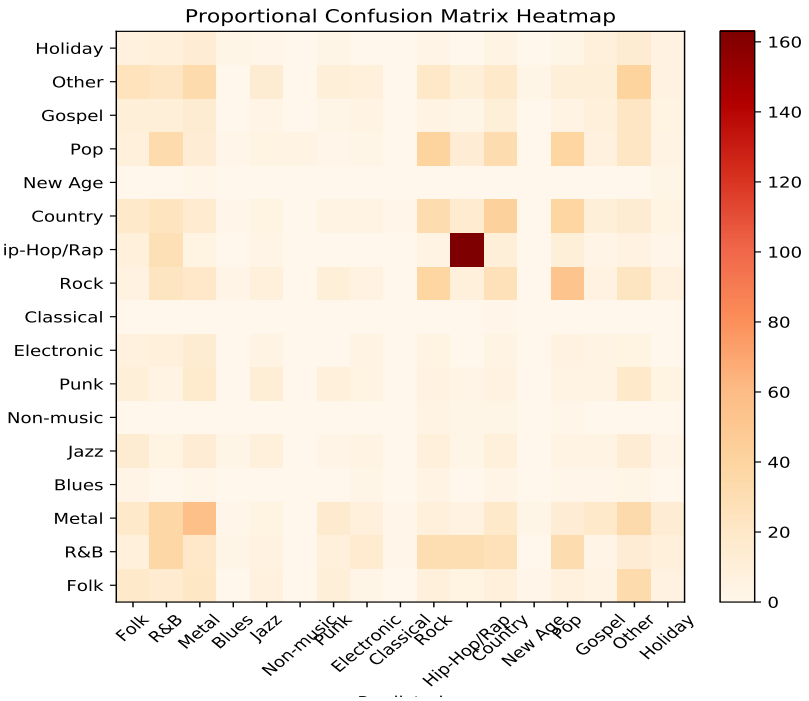

Figure 13: Confusion matrix heatmap for combined model

5 Topic Modelling

This task can be supervised but we do not have any training data (one would have had to manually tag documents with topics). We therefore, take an unsupervised approach that will be validated by sampling and manually evaluating the accuracy. We will compare two models: Latent-Dirichlet Allocation (LDA) and the doc2vec model from earlier.

5.1 Using LDA

LDA is common topic modelling technique that embed word-counts and document term matrices to identify a specified number of topics in a corpus. The model then finds this number of topics and describes each using a specified number of words. This is implemented for our dataset in the notebook "Topic Modelling with LDA.ipynb".

In the notebook, we assume 20 topics described by 5 words each. We then attempt to manually identify the overall concept of each topic. When we attempted to search for songs pertaining to specific topics (out of the list of 20), it did not prove to be very accurate but was able to identify some songs about specific ideas such as Christmas or religion.

The downsides to this method is that it can be very slow and the number of topics must be predefined. Once the list of topics is built, one can find documents pertaining to those topics, but the user has no control over the topics that the model will find. We therefore propose the following method which allows for the search for a specific topic or concept.

5.2 Using doc2vec

Doc2vec can be used to identify the topic of a document or search for a document that pertains to a given topic. For example, after training the model, one could search for a list of documents in the corpus that talk about LOVE or HAPPINESS.

Topic identification Like we did with genres, we can create "topic vectors" in a similar manner. These vectors are calculated from a set of words that describe a given topic. For example, the vector for "LOVE" might consist of the word vectors [*love, loving, heart, adore,...*]. The model then identifies lyrics that are close to this vector to be about LOVE. Again, we have not pre-tagged the documents with topics so there is no way to judge the accuracy of these predictions other than to manually review a sample. Figure 14 shows some topic vectors in the document space.

Concept search Using the same principle, we can search for songs about LOVE or SADNESS by creating a concept vector and searching for close documents. This is implemented in the notebook *Topic Modelling with doc2vec.py*

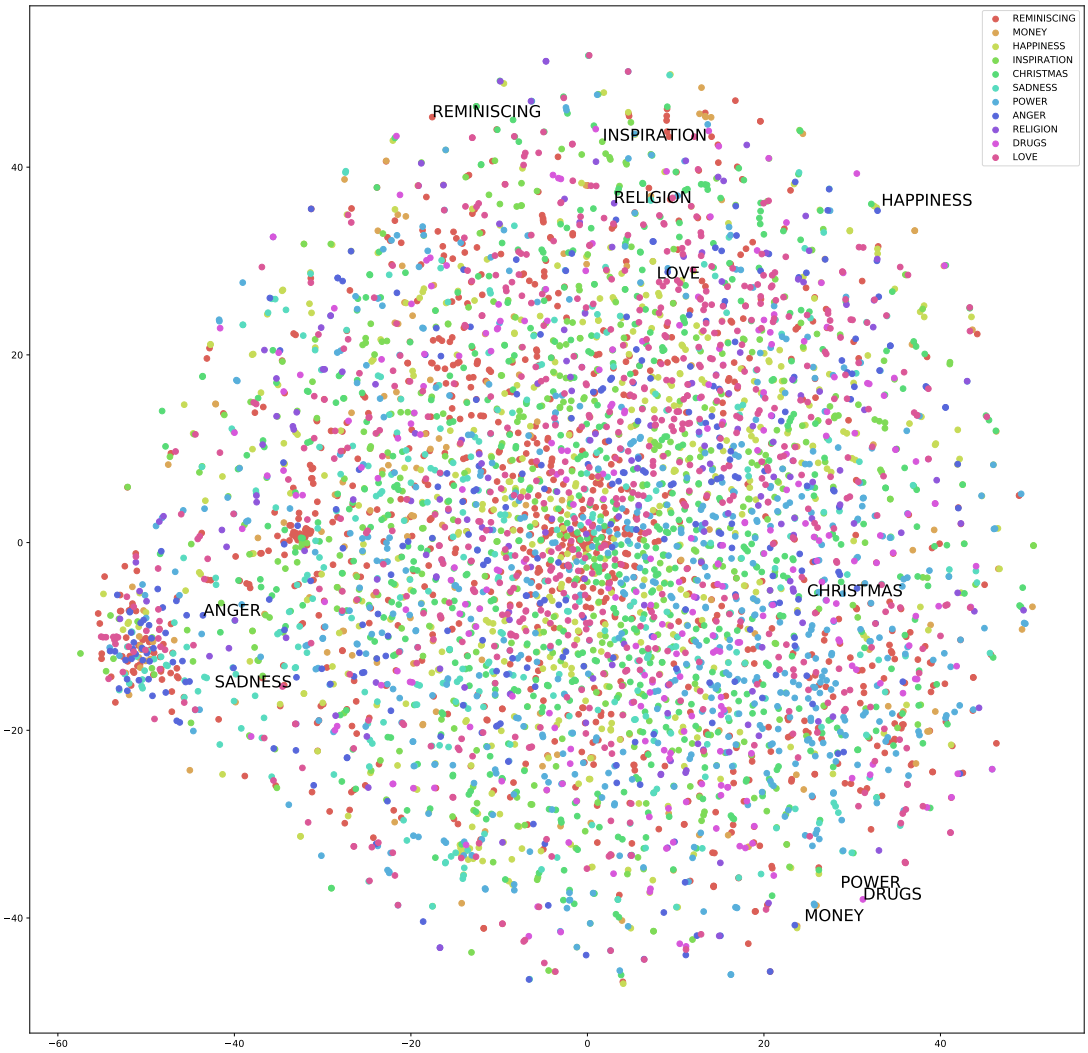

Figure 14: Topic vector in the document space (reduced to 2D for visualization)

and proves to perform well, in general. There is no metric by which to judge this model so it is up to the user to validate the results.

This model could be used to create a playlist of songs about a specific topic. It could also provide recommendations to a user. For example, a user could request a selection of love songs if they were in such a mood.

6 Conclusion

In conclusion, genre is a complicated attribute and classification is more difficult that we had thought. There are many factors that influence the creation and definition of a genre. Therefore, as we have seen from this analysis, the lyrics alone only represent a small part of the genre. However, we do conclude that some genres such as rap can be easily distinguished from other genres due to their unique lyric style.

In this paper, we have demonstrated the use of Word2vec to classify songs using their lyrics. The model can be applied to create playlists of similar songs or pertaining to a particular theme. This also provides a pipeline for such a model which could be run on larger datasets to improve results.

References

[1] Genius API.
https://docs.genius.com/

[2] Merriam Webster Inc. *genre*
https://www.merriam-webster.com/dictionary/genre

[3] 91.7 KOOP FM. *Genres & Definitions.*
http://www.koop.org/library/genres-definitions

[4] Yang Gao, John Harden, Vojtech Hrdinka, Chris Linn. *Lyric Complexity and Song Popularity: Analysis of Lyric Composition and Relation among Billboard Top 100 Songs.*
https://support.sas.com/resources/papers/proceedings16/11500-2016.pdf

[5] Colin Morris. *Are Pop Lyrics Getting More Repetitive?.*
https://pudding.cool/2017/05/song-repetition/

[6] Ali Doganaksoy, Faruk Gologlu. *On Lempel-Ziv Complexity of Sequences.*
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.8931&rep=rep1&type=pdf

[7] Swear Word List & Curse Filter.
https://www.noswearing.com/dictionary

[8] musicmap. *Abstract: The Definition of Genre* 2016.
https://musicmap.info/

[9] Google Archive: *word2vec*
https://code.google.com/archive/p/word2vec/

[10] Quoc Le, Tomas Mikolov. *Distributed Representations of Sentences and Documents.*
https://arxiv.org/pdf/1405.4053v2.pdf

[11] Christian S. Perone. *Machine Learning :: Cosine Similarity for Vector Space Models (Part III)*
http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/

[12] Radim Rehurek. *Gensim: models.doc2vec – Deep learning with paragraph2vec.*
https://radimrehurek.com/gensim/models/doc2vec.html

[13] Laurens van der Maaten. *t-SNE.*
https://lvdmaaten.github.io/tsne/

[14] *Doc2Vec Tutorial on the Lee Dataset.*
`https://github.com/RaRe-Technologies/gensim/blob/develop/docs/notebooks/doc2vec-lee.ipynb`

[15] Michael Fell, Caroline Sporleder. *Lyrics-based Analysis and Classification of Music.*
`http://www.aclweb.org/anthology/C14-1059`

Appendices

Appendix A Genre space

The following is a breakdown of the lyric space by genre.